

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS
CURSOS

FORMACIONES

CURSOS

PARA
EMPRESAS

ARTÍCULOS DE TECNOLOGÍA > DEVOPS

Entendiendo los permisos en el Linux



Yuri Matheus

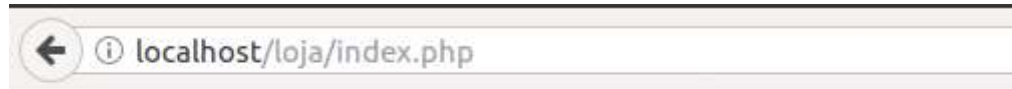
04/03/2022



Esse artigo faz parte da
Formação DevOps



Estoy trabajando en un proyecto **PHP**. Solo que cuando intento entrar por el navegador veo el siguiente error:



Forbidden

You don't have permission to access /loja/index.php on this server.

Apache/2.4.18 (Ubuntu) Server at localhost Port 80

Los archivos están hospedados en mi maquina local y así mismo no tengo acceso a él por el navegador. ¿Qué es lo que puede estar causando eso?

Si analizamos el mensaje de error, conseguimos obtener algunas informaciones como:

Prohibido, tú no tienes permiso para acceder loja/index.php en este servidor.

¿Pero cómo así no tenemos permiso? necesitamos, antes, entender lo que son esos permisos...

Entendiendo los permisos en Linux

En Linux, cuando un archivo o directorio es creado, algunos permisos son atribuidos a él automáticamente.

Esos permisos son divididos en tres grupos: El usuario que creo el archivo (usuario dueño), el grupo dueño del archivo (que puede contener varios usuarios), y de los demás usuarios que no pertenecen al grupo dueño.

Esos permisos pueden ser de lectura (r, read), escritura (w, write) y ejecución (x, execute).

El permiso de lectura (r) permite visualizar el contenido de un archivo o directorio, ya el permiso de escritura (w) sirve para alterar el contenido de un archivo y directorio.

La opción de ejecución sirve para tornar el archivo en ejecutable, o, en caso de directorios, para poder [accederlo](#) y usar comando, como el cd.

Cierto, ya sabemos lo que son y cuáles son los permisos, ¿pero como podemos saber cuáles están siendo usados en este archivo?

Visualizando los permisos en Linux

Para visualizar el permiso de un directorio, podemos ir hasta el local donde ese directorio se encuentra.

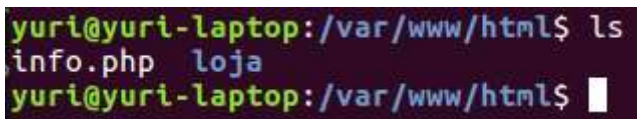
El comando `cd` (change directory) puede llevarnos hasta ese local, que en mi caso queda en `/var/www/html`. Por tanto:

```
$ cd /var/www/html/
```



```
yuri@yuri-laptop: /var/www/html
yuri@yuri-laptop:~$ cd /var/www/html/
yuri@yuri-laptop:/var/www/html$
```

Ahora podemos ordenarle a el terminal listar (`ls`, list) el contenido de este directorio:



```
yuri@yuri-laptop:/var/www/html$ ls
info.php  loja
yuri@yuri-laptop:/var/www/html$
```

Hum... Aun no conseguimos visualizar los permisos.

La lista que retorna el comando `ls` es una lista normal. Para visualizar los archivos y sus permisos tenemos que indicarle al `ls` que queremos una lista larga (`-l`).

```
$ ls -l
```

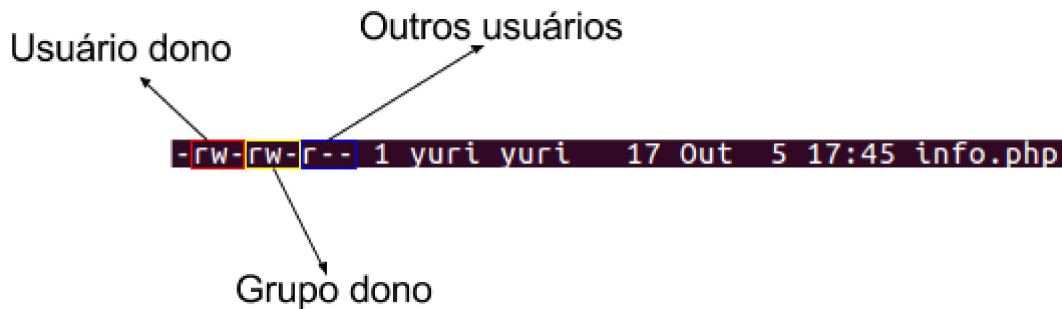


```
yuri@yuri-laptop:/var/www/html$ ls -l
total 8
-rw-rw-r-- 1 yuri yuri  17 Out  5 17:45 info.php
drwx----- 5 yuri yuri 4096 Feb 15 2017 loja
yuri@yuri-laptop:/var/www/html$
```

Ahora tenemos mas informaciones, pero, como queremos saber apenas de los permisos, vamos a concentrarnos en la primera columna.

La primera columna es dividida en diez caracteres. El primero nos muestra si el objeto listado es un archivo (`-`) o un directorio (`d`).

Los nueve caracteres restantes nos muestran los permisos de los usuarios dueños, grupo dueño y otros usuarios respectivamente:



En caso tenga algún permiso, es mostrada la letra correspondiente al permiso, si no un trazo (-) es presentado.

Sabiendo eso, ya conseguimos decir cuál es el permiso de nuestro proyecto loja. El usuario dueño tiene todos los permisos (rwx), mientras el grupo dueño y los demás usuarios no poseen ningún permiso (---).

Pero, si yo cree los archivos y tengo todos los permisos, ¿por qué no puedo entrar a la página por el navegador?

Cuando entramos a una página por el navegador, por defecto, el servidor no sabe quién es el usuario, o sea, quién entra a la página web utiliza el permiso **otro**.

Como el directorio del proyecto no posee permisos disponibles para el grupo de usuarios **otros** no conseguimos entrar a la página por el navegador.

Pues, necesitamos alterar los permisos de ese directorio para conseguir tener acceso por el navegador. ¿Pero cómo podemos hacer eso?

Alterando permisos en Linux

Para alterar los permisos de un archivo o directorio, tenemos que alterar el modo (chmod, change mode) que ellas están organizadas.

Queremos que los otros usuarios (o) tengan acceso de lectura (r) en el directorio, así podrán visualizar los archivos en el mismo. Entonces podemos digitar en el terminal:

```
$ chmod o=r loja/
```

```
yuri@yuri-laptop:/var/www/html$ chmod o=r loja/
yuri@yuri-laptop:/var/www/html$
```

Vamos a listar con la opción de lista larga para ver si ocurrió el cambio en los permisos:


```
yuri@yuri-laptop:/var/www/html$ ls -l
total 8
-rw-rw-r-- 1 yuri yuri 17 Out 5 17:45 info.php
drwx---r-- 5 yuri yuri 4096 Feb 15 2017 loja
yuri@yuri-laptop:/var/www/html$
```

¡Permiso atribuido! Vamos a revisar si ahora conseguimos nuestra página por el navegador:



Aun no obtuvimos éxito. Nosotros atribuimos la opción para la lectura, que en caso de directorios sirve para poder visualizar su contenido. Pero no tenemos permiso para introducir esos directorios. Para eso, necesitamos del permiso de ejecución (x).

Ya sabemos cómo alterar el permiso de una carpeta, entonces basta decir que queremos adicionar (+), el permiso de ejecución (x):

```
$ chmod o+x loja/
```

```
yuri@yuri-laptop:/var/www/html$ chmod o+x loja/
yuri@yuri-laptop:/var/www/html$
```

Vamos a intentar entrar nuevamente a nuestro proyecto por el navegador y ver si podemos visualizar nuestra página:



The screenshot shows a web browser window with the address bar displaying 'localhost/loja/index.php'. Below the address bar, the page title is 'Minha Loja'. There is a navigation menu with three links: 'Adiciona Produto', 'Produtos', and 'Sobre'. The main heading is 'Bem vindo!' (Welcome!). Below this is a 'Login' section with two input fields: 'Email' and 'Senha' (Password). A 'Login' button is positioned below the password field.

Bien la página fue cargada, ¿pero dónde está el CSS? ¿Por qué será que él no fue cargado?

Vamos a revisar los permisos de los archivos del proyecto y ver si tenemos permiso para introducirlos.

Para eso, podemos entrar en el directorio como hicimos antes con el comando `cd`, o entonces, podemos pedir para el `ls` listar el contenido del directorio tienda:

```
$ ls -l loja/
```

```
yuri@yuri-laptop:/var/www/html$ ls -l loja/
total 84
-rwx----- 1 yuri yuri 730 Feb 15 2017 adiciona-produto.php
-rwx----- 1 yuri yuri 699 Feb 15 2017 altera-produto.php
-rwx----- 1 yuri yuri 324 Feb 15 2017 banco-categoria.php
-rwx----- 1 yuri yuri 1305 Feb 15 2017 banco-produto.php
-rwx----- 1 yuri yuri 378 Feb 15 2017 banco-usuario.php
-rwx----- 1 yuri yuri 981 Feb 15 2017 cabecalho.php
-rwx----- 1 yuri yuri 66 Feb 15 2017 conecta.php
drwx----- 2 yuri yuri 4096 Feb 10 2017 css
drwx----- 2 yuri yuri 4096 Jul 25 2016 fonts
-rwx----- 1 yuri yuri 858 Feb 15 2017 index.php
drwx----- 2 yuri yuri 4096 Jul 25 2016 js
-rwx----- 1 yuri yuri 506 Feb 14 2017 logica-usuario.php
-rwx----- 1 yuri yuri 419 Feb 15 2017 login.php
-rwx----- 1 yuri yuri 140 Feb 15 2017 logout.php
-rwx----- 1 yuri yuri 204 Feb 14 2017 mostra-alerta.php
-rwx----- 1 yuri yuri 680 Feb 15 2017 produto-altera-formulario.php
-rwx----- 1 yuri yuri 1023 Feb 15 2017 produto-formulario-base.php
-rwx----- 1 yuri yuri 631 Feb 15 2017 produto-formulario.php
-rwx----- 1 yuri yuri 900 Feb 15 2017 produto-lista.php
-rwx----- 1 yuri yuri 276 Feb 15 2017 remove-produto.php
-rwx----- 1 yuri yuri 42 Feb 10 2017 rodape.php
yuri@yuri-laptop:/var/www/html$
```

Vea que ni los archivos o directorios tienen permiso para otros usuarios. O sea, no conseguimos introducir el contenido de los directorios. Para resolver esos problemas basta alterar el permiso de todas las carpetas y directorios.

“Pero, ¿tenemos que alterar el permiso de todos los archivos, directorios y subdirectorios uno por uno?”

¡No necesitamos! El `chmod` tiene una opción que hace eso para nosotros. Esa opción cambia el permiso de los archivos recursivamente. Esto es, ella cambia el permiso de todos los archivos en todo el directorio y subdirectorios.

Vamos entonces a indicarle al comando `chmod` cambiar el permiso de manera recursiva (`-R`) en el directorio `loja`:

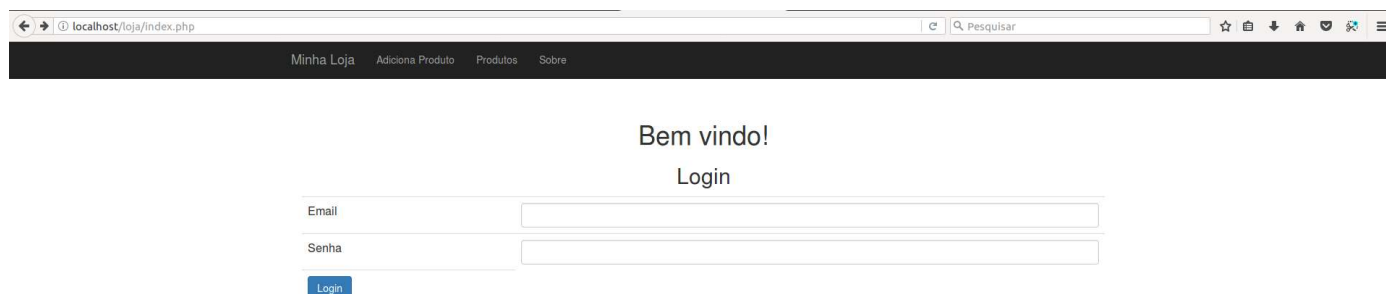
```
$ chmod -R o=rx loja/
```

```
yuri@yuri-laptop:/var/www/html$ chmod -R o=rx loja/
yuri@yuri-laptop:/var/www/html$
```

Vamos a revisar con el comando `ls` si el cambio ocurrió de hecho:


```
yuri@yuri-laptop:/var/www/html$ ls -l loja/
total 84
-rwx---r-x 1 yuri yuri 730 Feb 15 2017 adiciona-produto.php
-rwx---r-x 1 yuri yuri 699 Feb 15 2017 altera-produto.php
-rwx---r-x 1 yuri yuri 324 Feb 15 2017 banco-categoria.php
-rwx---r-x 1 yuri yuri 1305 Feb 15 2017 banco-produto.php
-rwx---r-x 1 yuri yuri 378 Feb 15 2017 banco-usuario.php
-rwx---r-x 1 yuri yuri 981 Feb 15 2017 cabecalho.php
-rwx---r-x 1 yuri yuri 66 Feb 15 2017 conecta.php
drwx---r-x 2 yuri yuri 4096 Feb 10 2017 css
drwx---r-x 2 yuri yuri 4096 Jul 25 2016 fonts
-rwx---r-x 1 yuri yuri 858 Feb 15 2017 index.php
drwx---r-x 2 yuri yuri 4096 Jul 25 2016 js
-rwx---r-x 1 yuri yuri 506 Feb 14 2017 logica-usuario.php
-rwx---r-x 1 yuri yuri 419 Feb 15 2017 login.php
-rwx---r-x 1 yuri yuri 140 Feb 15 2017 logout.php
-rwx---r-x 1 yuri yuri 204 Feb 14 2017 mostra-alerta.php
-rwx---r-x 1 yuri yuri 680 Feb 15 2017 produto-altera-formulario.php
-rwx---r-x 1 yuri yuri 1023 Feb 15 2017 produto-formulario-base.php
-rwx---r-x 1 yuri yuri 631 Feb 15 2017 produto-formulario.php
-rwx---r-x 1 yuri yuri 900 Feb 15 2017 produto-lista.php
-rwx---r-x 1 yuri yuri 276 Feb 15 2017 remove-produto.php
-rwx---r-x 1 yuri yuri 42 Feb 10 2017 rodape.php
yuri@yuri-laptop:/var/www/html$
```

¡Todo listo! Ahora si entramos por el navegador devemos ver el site completo:



¡Éxito! el site cargó, así como todos sus componentes.

Conseguimos alterar los permisos de nuestros archivos, pero solo usamos la opción para otros usuarios (o). ¿Qué hacer en caso de querer alterar los permisos del usuario y del grupo dueño del archivo?

Alterando el permiso del usuario y del grupo dueño en Linux

Antes de mostrar como alterar el permiso del usuario el grupo dueño de un archivo o directorio, vamos crear un archivo de texto en cualquier lugar del sistema para ser usado como prueba. En mi caso voy crear ese archivo en el área de trabajo:


```
yuri@yuri-laptop:~/Área de Trabalho$ ls -l
total 0
-rw-rw-r-- 1 yuri yuri 0 Out 24 17:06 teste.txt
yuri@yuri-laptop:~/Área de Trabalho$
```

Ese archivo ya contiene algunos permisos que fueron atribuidos por defecto en el momento en que fué creado.

Vamos a ordenarle con el comando `chmod` que queremos que el usuario dueño del archivo (u) tenga, además de esos permisos, el permiso de ejecutar el archivo.

Entonces yo puedo ordenarle al comando `chmod` aumentar (+) el permiso de ejecución para el usuario dueño (u):

```
$ chmod u+x teste.txt
```

```
yuri@yuri-laptop:~/Área de Trabalho$ chmod u+x teste.txt
yuri@yuri-laptop:~/Área de Trabalho$
```

También podemos ordenarle al comando `chmod` retirar (-) el permiso de escritura del grupo dueño (g):

```
$ chmod g-w teste.txt
```

```
yuri@yuri-laptop:~/Área de Trabalho$ chmod g-w teste.txt
yuri@yuri-laptop:~/Área de Trabalho$
```

Si le ordenamos listar ese archivo y sus permisos, podemos ver los cambios:

```
$ ls -l
```

```
yuri@yuri-laptop:~/Área de Trabalho$ ls -l
total 0
-rwxr--r-- 1 yuri yuri 0 Out 24 17:06 teste.txt
yuri@yuri-laptop:~/Área de Trabalho$
```

"Hum... pero si yo quisiera cambiar el permiso de todos los usuarios en un único comando, ¿cómo puedo hacer?"

Nosotros podemos, por ejemplo, pedirle al `chmod` remover el permiso de ejecución del usuario dueño del archivo (u-x), agregar las opciones de escritura y ejecución para el grupo dueño (g+wx) y dejar los demás usuarios apenas con el permiso de ejecución (o=x):

```
$ chmod u-x,g+wx,o=x teste.txt
```

```
yuri@yuri-laptop:~/Área de Trabalho$ chmod u-x,g+wx,o=x teste.txt
yuri@yuri-laptop:~/Área de Trabalho$
```

¡Listo! Alteramos todos los permisos en apenas una línea con un único comando. Si visualizamos los permisos ahora, veremos los cambios:

```
yuri@yuri-laptop:~/Área de Trabalho$ ls -l
total 0
-rw-rwx--x 1 yuri yuri 0 Out 24 17:06 teste.txt
yuri@yuri-laptop:~/Área de Trabalho$
```

Pero necesitamos digitar mucho para alterar todos los permisos. ¿Existe alguna forma de ser más directo en la digitación?

Además de utilizar letras, llamado de modo simbólico, o UGO (User, Group, Others), el `chmod` acepta también algunos números. Cuando utilizamos números para alterar los permisos, decimos que estamos utilizando el **modo octal**.

Entendiendo el modo octal

El modo octal recibe este nombre, pues utilizamos ocho números, de 0 a 7, cada uno de esos números corresponden a una letra, o a un conjunto de letras, en el modo simbólico:

- 1 → Representa la opción de ejecución (x) en el modo simbólico;
- 2 → La opción de escritura (w);
- 4 → La opción de lectura (r).

Cuando utilizamos el modo octal, podemos pasar el modo de permisos de cada grupo de usuarios. La orden es siempre: usuario dueño, grupo dueño y otros usuarios.

Entonces, por ejemplo yo ordenarle al comando `chmod`, colocar el permiso de lectura para el usuario dueño (4), para el grupo dueño de la escritura (2) y la de ejecución para los demás usuarios (1):

```
$ chmod 421 teste.txt
```

```
yuri@yuri-laptop:~/Área de Trabalho$ chmod 421 teste.txt
yuri@yuri-laptop:~/Área de Trabalho$
```

Ese comando equivale a `chmod u=r,g=w,o=x teste.txt` en modo simbólico.

¿Solo podemos utilizar esas opciones con el modo octal?

Nosotros también conseguimos combinar esos permisos en el modo octal. Por ejemplo, si deseamos que el usuario dueño del archivo tenga el permiso de lectura (4) y escritura (2), basta sumar el valor de esos permisos. Por tanto, tendríamos 6.

Ya si queremos quitar todos los permisos, basta colocar 0.

Para que el usuario dueño tenga todos los permisos (7) en el archivo, y el grupo dueño tenga los permisos de lectura y escritura (6) y los demás usuarios no tengan ningún permiso (0). Basta indicarle al comando `chmod`:

```
$ chmod 760 teste.txt
```

```
yuri@yuri-laptop:~/Área de Trabalho$ chmod 760 teste.txt  
yuri@yuri-laptop:~/Área de Trabalho$
```

Si listamos los permisos, podemos revisar si los cambios funcionaran:

```
yuri@yuri-laptop:~/Área de Trabalho$ ls -l  
total 0  
-rwxrw---- 1 yuri yuri 0 Out 24 17:06 teste.txt  
yuri@yuri-laptop:~/Área de Trabalho$
```

Nuestros permisos fueron atribuidos correctamente. :)

Para saber más sobre permisos

Además del `chmod`, otro comando muy utilizado cuando es el **chown** (change owner). Ese comando consigue cambiar el usuario, el grupo dueño de un archivo o directorio y solo puede ser ejecutado por el super usuario, o por el usuario administrador utilizando el comando `sudo`.

Debemos tener mucho cuidado en dar permisos para los usuarios que no son parte del grupo dueño, o no son dueños de los archivos o carpetas. Pues, dependiendo del permiso, esos usuarios consiguen tener acceso a un directorio o archivo en el sistema. Pudiendo visualizar, alterar, o hasta mismo excluir su contenido.

¿Tiene permiso de continuar?

Permisos hacen parte del sistema. Sin ellos no podemos visualizar, alterar o ejecutar archivos, entrar en directorios o listar su contenido. Hasta mismo para acceder a un sitio en el internet necesitamos de permisos.

En Linux, podemos alterar el permiso de archivos y directorios usando un comando propio para eso, el `chmod`. Ese comando nos permite alterar los permisos del usuario y grupo dueño del archivo o directorio y de los demás usuarios.

Pero antes de alterar, tenemos que entender cuáles son esos permisos y como pueden ser representados.



Yuri Matheus

Yuri es desarrollador e instructor. Es estudiante de Sistemas de Información en la FIAP y es formado como Técnico en Informática en el Senac SP. Y su foco es en las plataformas Java y Python y en otras áreas como Arquitectura de Software y Machine Learning. Yuri también actúa como editor de contenido en el blog de Alura, donde escribe, principalmente, sobre Red, Docker, Linux, Java y Python.



Esse artigo faz parte da
Formação DevOps

Cursos de DevOps

ARTÍCULOS DE TECNOLOGÍA > DEVOPS

En Alura encontrarás variados cursos sobre DevOps. ¡Comienza ahora!

SEMESTRAL

US\$49,90

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

ANUAL

US\$79,90

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

Acceso a todos
los cursos

Estudia las 24 horas,
dónde y cuándo quieras

Nuevos cursos
cada semana

NAVEGACIÓN

PLANES

INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A

CNPJ 05.555.382/0001-33

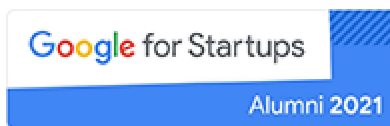
SÍGUENOS EN NUESTRAS REDES SOCIALES



ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

CURSOS

Cursos de Programación

Lógica de Programación | Java

Cursos de Front End

HTML y CSS | JavaScript | React

Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

Cursos de DevOps

Docker | Linux

Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics |

Liderazgo y Gestión de Equipos | Startups y Emprendimiento